PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-140758
(43)Date of publication of application : 16.05.2003

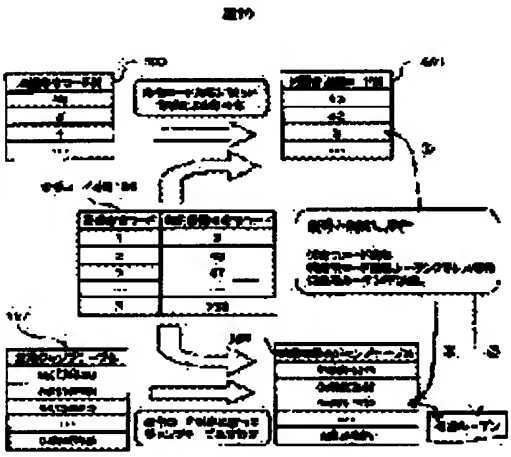
(51)Int.Cl. G06F 1/00

(21)Application number : 2001-341368
(22)Date of filing : 07.11.2001

(71)Applicant : HITACHI LTD
(72)Inventor : KAWASAKI SHINICHIRO
SATOYAMA MOTOAKI
YOKOYAMA YASUKO
MORIMOTO YOSHIAKI
KITAGAWA KENJI

(54) PROGRAM ENCIPHERING/DECIPHERING METHOD AND ITS EXECUTING METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To solve the problem that it takes a long time to decode a distribution program and it is necessary to secure a storage region for decoding the program at the time of distributing a program through a network by using a conventional enciphering method.
SOLUTION: A method for exchanging the correspondence of an 'instruction code' held by a program to 'instruction contents' is used as a method for enciphering the program. Thus, it is possible to realize the interpretation and execution of the program simultaneously with the decoding of the program. Therefore, it is possible to shorten the decoding time, and it is not necessary to secure any storage region for decoding.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2003－140758
（P2003－140758A）
(43)公開日　平成15年５月16日(2003.5.16)

(51)Int.Cl.[7]	識別記号	ＦＩ	テーマコード（参考）
Ｇ０６Ｆ　1/00		Ｇ０６Ｆ　9/06	６６０Ｄ　５Ｂ０７６

審査請求　未請求　請求項の数４　ＯＬ　（全 11 頁）

(21)出願番号　特願2001－341368(P2001－341368)

(22)出願日　平成13年11月７日(2001.11.7)

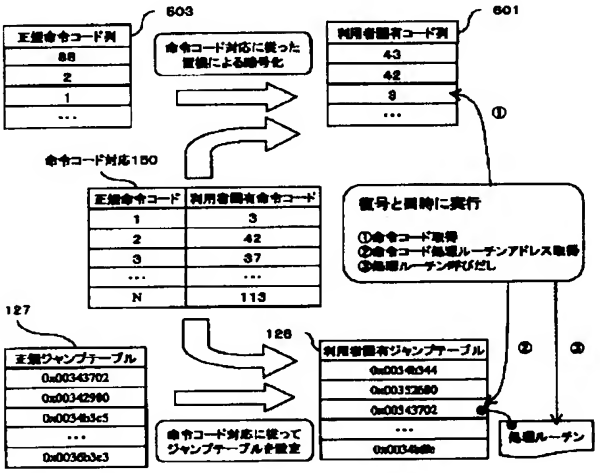
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　川▲崎▼　進一郎
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　里山　元章
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】　プログラム暗号化方法、復号化方法および実行方法

(57)【要約】
【課題】ネットワークを通じてプログラムを配信する時などに、従来の暗号化手法を用いると配信プログラムの復号に長い時間がかかり、なおかつ復号のための記憶領域を必要とした。
【解決手段】プログラムの暗号化方法として、プログラムの保持する「命令コード」と「命令内容」の対応付けを入れ換える方法を用いる。これにより復号と同時にプログラムを解釈・実行することが可能となり、復号時間を短縮でき、しかも復号のための記憶領域が不要となる。

図10



1

【特許請求の範囲】

【請求項１】命令を表す数値の列として表現されたプログラムを暗号化する方法であって、
置換前の命令数値を置換後の数値に対応付ける対応表を用いて、命令数値を置換することを特徴とするプログラム暗号化方法。

【請求項２】請求項１記載の暗号化方法により暗号化されたプログラムを復号化すると同時に解釈実行する方法であって、
各命令を実行するルーチン各々のメモリアドレスを、置換後の命令数値の並びに従って整列させたアドレス表を、前記対応表に従って作成し、
上記アドレス表を参照することにより各命令の復号化と解釈実行を同時に行なうことを特徴とするプログラム復号化方法。

【請求項３】ネットワーク接続された配布サーバと中間コード実行装置において、配布サーバから中間コード実行装置に配布されたプログラムを実行するプログラム実行方法であって、
配布サーバが請求項１記載の暗号化方法によってプログラムを暗号化してから中間コード実行装置に配布し、
中間コード実行装置は請求項２記載の方法に従って復号化・実行することを特徴とし、
配布サーバ装置上にて請求項１記載の対応表を生成し、
配布サーバ装置から中間コード実行装置に上記対応表を配布し、
中間コード実行装置は上記対応表を参照して請求項２記載のアドレス表を設定することを特徴とするプログラム実行方法。

【請求項４】請求項２記載のプログラム実行方法において、請求項２記載のアドレス表を複数備え、暗号化されたプログラムと、暗号化されていないプログラムを単一の中間コード実行環境において同時に実行するために、実行中のプログラムが切替わる時に使用するアドレス表を切替えることを特徴とするプログラム実行方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、暗号化されたプログラムを実行する技術に関し、特にネットワークなどを通じて配布される、暗号化されたプログラムを実行する技術に関する。

【０００２】

【従来の技術】インターネットを初めとするネットワーク技術の進歩に伴い、ネットワークに接続された中間コード実行装置にプログラムを配信するサービスが実現されている。特に最近ではSun microsystems社のJava技術などの出現によって中間コードを用いてプログラムが記述されるようになり、複数種類の中間コード実行装置に対して同一のプログラムを配信できるようになった。

【０００３】これらのプログラム配布においては、配布

2

対象を特定の利用者に限定したい場合が多い。しかし、配布にはインターネットのような公共ネットワークが用いられることから、配布ミスや盗聴によって第三者の手にプログラムが渡ってしまう可能性があり、なんらかの対策が必要である。

【０００４】対策の一つとしてプログラムを暗号化して送信する方法がある。特開平10-161864では、プログラムの一部を暗号化することで、プログラム利用者によるコピーを含む違法コピーを防ぐ方法が示されている。プログラムを受信した中間コード実行装置は、まず暗号鍵を用いてプログラムを復号し、復号したプログラムを実行する。暗号鍵を持たない中間コード実行装置はプログラムを復号することができない。よって、プログラム実行は暗号鍵を持つ中間コード実行装置に限定される。

【０００５】

【発明が解決しようとする課題】上記従来技術は、暗号化されたプログラムを「復号」してから「実行」するという二段階のステップを踏んでいるため、復号処理のためにプログラム起動時間が長くなるという問題と、復号処理のための記憶領域を必要とするという問題があった。

【０００６】特に、中間コードによるプログラム配信サービスにおいては、中間コードに対する事前処理も必要であるため、プログラム受信から起動までにかかる時間が長いものになっていた。

【０００７】また、記憶領域の少ない中間コード実行装置においては、復号処理のための記憶領域の確保が難しいという問題があった。

【０００８】

【課題を解決するための手段】上記の問題点を解決するために、中間コードの命令対応付けに従って処理ルーチンへのジャンプ先アドレステーブルを変更する「命令処理ルーチンジャンプテーブル変更手段」と、中間コードを一つずつ取得し、ジャンプテーブルを用いて処理ルーチンを呼び出す「中間コード解釈実行手段」を備えた中間コード実行装置を設ける。これにより、プログラム起動時の復号処理にかかる時間が省略され、復号のための記憶領域が不要となる。

【０００９】

【発明の実施の形態】以下、本発明についての実施例の一つを、図１から図１７を用いて説明する。

【００１０】図１により、本発明を実現するシステムの全体構成を示す。

【００１１】本システムは、ネットワークにより接続された配布サーバ装置100と、複数の中間コード実行装置120(120a～120c...)などから構成される。

【００１２】配布サーバ装置100はネットワークを通じて中間コード実行装置120にプログラムを配布し、サービスを提供する。

【００１３】このシステムの応用例として、携帯電話へ

のプログラム配布サービスが挙げられる。携帯電話が中間コード実行装置120に相当し、携帯電話用のプログラムを格納したウェブサーバなどが配布サーバ装置100に相当する。

【００１４】配布サーバ装置100は主記憶装置101と二次記憶装置102と中央処理装置103と通信装置104などから成る。主記憶装置101は、命令コード対応管理モジュール105とプログラム管理モジュール106などを保持する。中央処理装置103は主記憶装置101上のモジュール群を動作させる。通信装置104は、ネットワークと通じて中間コード実行装置120と通信する機能を提供する。

【００１５】二次記憶装置102上には、利用者リスト107、命令コード対応リスト108、正規プログラムデータ109などが格納されており、主記憶装置101上のモジュールから読み書きされる。

【００１６】命令コード対応管理モジュール105は、利用者リスト107からサービス利用者の情報を取得し、利用者毎に一意の「命令コード対応150」を生成する機能と、生成された命令コード対応150を当該利用者が利用している中間コード実行装置120に送信する機能を実現する。

【００１７】プログラム管理モジュール106は、中間コード実行装置120からプログラム配布要求151を受信する機能と、配布するプログラムを暗号化する機能と、中間コード実行装置120に暗号化済プログラム152を送信する機能を実現する。

【００１８】中間コード実行装置120は、主記憶装置121、通信装置122、中央処理装置123などから構成される。主記憶装置121には、プログラム管理モジュール124、命令コード対応受信モジュール125、プログラム復号・実行モジュール126、正規ジャンプテーブル127、利用者固有ジャンプテーブル128などが配置される。中央処理装置123は主記憶装置121上の各モジュールを実行する。通信装置122は、ネットワークを通じて配布サーバ装置100と通信する機能を提供する。

【００１９】プログラム管理モジュール124は、配布サーバ100に対してプログラムの配布を要求する機能と、暗号化済みプログラム152を配布サーバ100から受信する機能と、受信したプログラムを保持する機能を実現する。

【００２０】命令コード対応受信モジュール125は配布サーバ装置100から命令コード対応150を受信する機能と、その命令コード対応150を用いて利用者固有ジャンプテーブル128を書き換える機能を実現する。

【００２１】プログラム復号・実行モジュール126は、利用者固有ジャンプテーブル128を用いて、暗号化済みプログラム150の復号と実行を同時に行なう機能を実現する。正規命令処理ジャンプテーブル127は、プログラム復号・実行モジュール126が暗号化されていない正規プログラムを実行する時に参照される。各ジャンプテーブルは、各中間コード命令の処理ルーチンアドレスを保持している。詳細は図７・図８を用いて後述する。

【００２２】以下、図２から図９を用いてシステム構成の詳細を示す。

【００２３】図２を用いて利用者リスト107の一例を示す。利用者リストは、利用者ＩＤ202、氏名、中間コード実行装置ネットワークアドレスからなる利用者データを複数保持する。中間コード実行装置ネットワークアドレスは、当該利用者が利用する中間コード実行装置120を、ネットワーク上で一意に識別できるアドレスである。

【００２４】図３を用いて、中間コードの正規命令コード対応について説明する。

【００２５】一つのプログラムを異なるハードウェア上で動作させるために、ハードウェアに依存しない中立的な命令群を用いてプログラムを記述する。このための中立的な命令列が中間コードである。各ハードウェア上では、中間コードを解釈・実行する仮想機械を動作させる。図３に示したように、整数加算、整数引き算などの各命令に、１、２などの数値を対応づけることで、正規命令のコードが定まる。この対応付けの個数は、定義された中間コードの命令数Ｎとなる。

【００２６】図４を用いて、命令コード対応リスト108の一例を示す。命令コード対応リスト108は、利用者ＩＤ202と命令コード対応150のリストを保持する。命令コード対応150は、中間コードの正規命令数値と、それに対応づけられた数値のペアである利用者固有命令コードを保持する。

【００２７】図５に、正規プログラムデータ109のデータ構成の一例を示す。

【００２８】正規プログラムデータ109は、複数の正規プログラム501から構成される。正規プログラム501はプログラムＩＤ502、プログラム長などと一緒に保持される。正規プログラム501は、開発元情報、プログラム名、関数リストなどから構成される。関数リストは複数の関数データから構成される。関数データは、関数名、返り値型、引数型リスト、正規命令コード列503などから構成される。引数型リストは複数の引数名と引数型から構成される。正規命令コード列503は、図３に示した正規命令コード対応に従った、正規命令コードの列であり、当該関数の処理内容を記述するものである。従来の中間コード実行装置は、この正規命令コード列を解釈・実行する。

【００２９】図６に、暗号化済プログラム152のデータ構成の一例を示す。

【００３０】暗号化済プログラム152は図5に示した正規プログラム501とほぼ同様のデータ構成を持つ。正規プログラム501は、関数データ中に正規プログラム命令コード列503を保持していたのに対し、暗号化済プログラム152は、利用者独自命令コード列601を保持している点

が異なる。

【００３１】正規プログラムから、暗号化済プログラムを生成する手順は後述する。

【００３２】図７に、中間コード実行装置120上の、正規ジャンプテーブル127のデータ構造の一例を示す。

【００３３】正規ジャンプテーブル127は、正規命令コードの順番に、正規該当命令コードを処理するルーチンのメモリアドレスをＮ個並べたものである。図７は、整数加算を示す正規命令コード「１」に対応するメモリアドレスに、整数加算のルーチンが配置されている状況を示す。

【００３４】図８に、利用者固有ジャンプテーブル128のデータ構造の一例を示す。

【００３５】利用者固有ジャンプテーブル128は、正規ジャンプテーブル127と同様の構造を持っているが、正規命令コードの順番ではなく、利用者固有命令コードの順番に処理ルーチンアドレスを並べたものである。図８は整数加算を示す利用者固有命令コード「３」に対応するメモリアドレスに、整数加算のルーチンが配置されている状況を示す。

【００３６】図９に、プログラム配布要求151のデータ構造の一例を示す。

【００３７】プログラム配布要求151は、要求元を示す、中間コード実行装置ネットワークアドレスと、利用者ＩＤ202を保持し、配布を要求するプログラムを示すプログラムＩＤ 502を保持する。

【００３８】図１０を用いて、本発明の中心となるプログラムの暗号化処理と解釈実行処理の流れを説明する。

【００３９】最初に、正規命令コード列503に対し、命令コード対応150に従って命令を表す数値を置換することによりプログラムが暗号化される。図１０では、正規命令コードでは「１」であった命令が、置換によって「３」になっている。暗号化されたコード列は利用者固有コード列601となる。

【００４０】次に正規ジャンプテーブル127と命令コード対応150から、利用者固有ジャンプテーブル128を設定する。図７に示したように、正規ジャンプテーブルには正規命令コード順に各命令処理ルーチンのアドレスが並べられている。これを図８に示したように利用者固有命令コードの順番に並べなおす。正規ジャンプテーブル127では１番目に配置されていたアドレス「0x00343702」は、利用者固有ジャンプテーブルでは３番目に配置されている。

【００４１】利用者固有コードの実行は、以下の３つの手順を繰り返すことで実現される。

（１）利用者固有コードを一つ利用者固有コード列601から取得。

（２）取得したコードを処理する処理ルーチンのアドレスを、利用者固有ジャンプテーブルから取得。

（３）取得した処理ルーチンを呼び出す。

【００４２】上記手順は、正規ジャンプテーブル127の代わりに利用者固有ジャンプテーブル128を用いている点を除き、通常の中間コード解釈実行処理と同じ処理である。よって特別な命令コードの復号処理は不要であり、復号処理のための記憶領域も必要としない。

【００４３】以下、図１１と図１２を用いて、システム全体の処理手順を示す。本実施例の手順は、大きく「命令コード対応の生成・配布」と「プログラム配布・実行」に分かれる。「命令コード対応の生成・配布」処理は、「プログラム配布・実行」処理の事前に行なわれる。

【００４４】図１１に「命令コード対応の生成・配布」の手順を示す。

処理1101：　配布サーバ装置100上の、命令コード対応管理モジュール105により、各利用者固有の命令コード対応150が生成され、命令コード対応リスト108に格納される。処理1101の詳細は図１３を用いて後述する。

処理1102：　配布サーバ装置100上の命令コード対応管理モジュール105により、命令コード対応リスト108に格納された各利用者固有の命令コード対応150が、各中間コード実行装置120に送信される。

【００４５】命令コード対応150は、「プログラム配布・実行」処理時に暗号鍵の役割を果たすデータであり、配信時に他者に盗聴されてはならない。このことから、配信にあたっては十分信頼できる暗号処理を適用することが望ましい。本処理および処理1103は、「プログラム配布・実行」処理の事前に行なわれる処理であり、ここでの暗号化処理、復号処理に処理時間がかかっても「プログラム配布・実行」時の処理時間には影響が無い。よってここで用いる暗号・復号処理は既存の処理時間が長い処理を用いてもよい。

処理1103：　中間コード実行装置120上の、命令コード対応受信モジュール125により、命令コード対応150が受信される。命令コード対応150が暗号化されている場合には復号される。

【００４６】さらに受信した命令コード対応150を元に、利用者固有ジャンプテーブル128が変更される。処理1103の詳細は図１５を用いて後述する。

【００４７】本実施例ではネットワーク通信による命令コード対応の配布処理を示したが、配布処理形態に制限はない。電子メールによる配布、ＣＤ－ＲＯＭやフロッピー（登録商標）ディスクの郵送による配布なども考えられる。

【００４８】図１２に「プログラム配布・実行」の手順を示す。

処理1201：　中間コード実行装置120上のプログラム管理モジュール124により、プログラム配布要求151が配布サーバ装置100に送信される。

処理1202：　配布サーバ装置100上のプログラム管理モジュール106により、プログラム配布要求151が受信され

る。
処理1203：　配布サーバ装置100上のプログラム管理モジュール106により、プログラム配布要求151に従って配布するプログラム正規プログラムデータ109から取得され、暗号化される。詳細は図１６を用いて後述する。
処理1204：　配布サーバ装置100上のプログラム管理モジュール106により、暗号化済みプログラム152が中間コード実行装置120に送信される。
処理1205：　中間コード実行装置120上のプログラム管理モジュール124により、暗号化済みプログラム152が受信される。
処理1206：　中間コード実行装置120上のプログラム復号・実行モジュール126により、暗号化済みプログラム152が復号と同時に実行される。詳細は図１７を用いて後述する。
【００４９】以下、図１３から図１７を用いて、処理手順の詳細を示す。
【００５０】図１３に「命令コード対応生成」処理1101の手順詳細を示す。
処理1301：　利用者リスト107を参照し、格納されている利用者データ201を一つ選択する。
処理1302：　選択した利用者に固有の命令コード対応150を生成する。手順の詳細は図１４を用いて後述する。
処理1303：　生成した命令コード対応150を、命令コード対応リスト108に格納する。
処理1305：　利用者リスト107に格納されている全ての利用者データ201について、命令コード対応150を生成したかどうか判定する。まだ生成していない利用者データ201がある場合には、処理1301に戻り、全利用者について生成を完了した場合には「命令コード対応生成」処理を終了する。
【００５１】図１４に「利用者固有命令コード対応生成」処理1302の手順詳細を示す。
処理1401：　中間コードの正規命令Ｎ個の中から、対応付けを定めていない正規命令を一つ選択する。
処理1402：　前記処理1401において選択した正規命令に対応付ける数値を、適切な乱数アルゴリズムを用いて生成し、対応付けるコードとする。
処理1403：生成したコードが、他の正規命令に対応付けるコードとして使われているかどうかを判定する。もし使われている場合には処理1402に戻る。使われていなければ処理1404に進む。
処理1404：　全ての正規命令に対してコードを対応付けたかどうかを判定する。まだ対応付けていない正規命令がある場合には処理1401に戻る。全てに対応付けを終えた場合は処理1405に進む。
処理1405：　上記処理1401から処理1404によって生成した「利用者固有命令コード対応」について、他利用者に固有の命令コード対応150と同じ対応付けになっていないかどうかを判定する。
【００５２】他利用者の命令コード対応150は、命令コード対応リスト108に格納されている。もし同じ対応付けが見つかれば処理1406に進み、見つからなければ生成した対応付けを「利用者固有命令コード対応」処理1303に引き渡して終了する。
処理1406：　各正規命令に処理1402によって生成されて対応付けられたコードを無効化する。これにより、スタート時点と同じ状態となる。
【００５３】図１５に「命令コード対応受信」処理1103の処理詳細を示す。
処理1501：　配布サーバ100から、命令コード対応150を受信する。命令コード対応150が暗号化されている場合には復号する。
処理1502：　受信した命令コード対応150から、利用者固有命令コードをひとつ選択する。
処理1503：　上記処理1502において選択した利用者固有命令コードに対応する正規命令コードを、受信した命令コード対応150から取得する。
処理1504：　上記処理1503により取得した正規命令コードが指定する命令を処理するルーチンのメモリアドレスを、正規ジャンプテーブル127から取得する。
処理1505：　上記処理1504により取得した命令処理ルーチンのアドレスを、利用者固有ジャンプテーブル128に格納する。格納する場所は、処理1502において選択した利用者固有命令コードから定める。
処理1506：　上記処理1502から処理1505を、全利用者固有命令コードについて行なったかどうかを判定する。まだ全てについて行なっていない場合は処理1502に戻る。完了した場合は処理を終了する。
【００５４】図１６に「プログラム暗号化」処理の手順詳細を示す。
処理1601：　プログラム配布を要求した利用者の利用者ＩＤ202を調べ、当該利用者の利用者固有命令コード対応150を命令コード対応リスト108から取得する。
処理1602：　配布要求されたプログラムのプログラムＩＤ 502を用いて、正規プログラムデータ109から正規プログラム501を取得する。
処理1603：　正規プログラム501の保持する正規コード列503から、正規命令コードを一つ取得する。
処理1604：　上記処理1601において取得した利用者固有命令コード対応を用いて、上記処理1603において取得した正規命令コードに対応する利用者命令コードを取得する。
処理1605：　上記処理1604において取得した利用者命令コードを用いて、正規命令コード列503の該当コードを置換し、上書きする。
処理1606：　正規命令コード503の、全命令を置換したかどうか判定する。置換が完了していない場合には処理1603に戻り、完了している場合には処理1607に進む。
処理1607：　上記置換作業により正規プログラム501か

ら変換された暗号化済みプログラム152を、処理1204に引き渡して処理を終了する。

【００５５】図１７に「プログラム復号・実行」処理1206の手順詳細を示す。

処理1701：　暗号化済みプログラム152から、実行する利用者固有命令コード列601を取得する。

処理1702：　上記処理1701において取得した利用者固有命令コード列601から、次に実行する命令コードを取得する。

処理1703：　取得した命令コードを処理するルーチンのメモリアドレスを、利用者固有ジャンプテーブル128を参照して取得する。この処理が、利用者固有命令を正規命令に戻す処理と、該当する処理ルーチンを取得する処理を、同時に行なっている。受信した暗号化済みプログラムの暗号化に用いられた、「命令コード対応150」を事前に受信していない中間コード実行装置120は、本処理に用いる利用者固有ジャンプテーブル128が適切でない。このため適切な処理ルーチンのメモリアドレスを取得することができず、中間コードを正しく実行できない。

処理1704：　上記処理1703において取得したメモリアドレスにある命令処理ルーチンを呼び出す。

処理1705：　上記処理1704におけるルーチン呼び出しの結果からプログラム実行を終了すべきかどうか判定する。実行を継続する場合は処理1702に戻る。

【００５６】次に、本発明を実現するもう一つの実施例を、図１から図１６、図１８から図２０を用いて説明する。

【００５７】配布サーバ装置100は、前述の実施例と同様の構成を持つ。中間コード実行装置120も前述の実施例とほぼ同様の構成であるが、図１８に示すように基本プログラムデータ1801が主記憶装置121上に追加されている点が異なる。

【００５８】図１８に、中間コード実行装置120のもう一つの構成例を示す。基本プログラム1801には、頻繁に利用される基本的な正規プログラムが予め格納されている。本実施例では、中間コード実行装置120は受信した暗号化済プログラム152のみではなく、基本プログラムデータ1801に格納されている正規プログラム501をも実行する。このとき、これらのプログラムは互いに関数呼びだしなどの方式により、混在した形で実行される。

【００５９】上記のように、暗号化済プログラム152と正規プログラム501を混在させて実行するために、各命令処理ルーチンへのジャンプテーブルとして、利用者固有ジャンプテーブル128、および正規ジャンプテーブル127のふたつを切り替えながら解釈・実行を遂行する手段を取る。この手段により、上記二種類のプログラムを混在させて実行することが可能になる。実行処理の詳細は図２０を用いて後述する。

【００６０】図１９に、基本プログラムデータ1801のデータ構造例を示す。複数の正規プログラム501が、プログラムＩＤ、プログラム長などと共に保持されている。

【００６１】図２０に二つ目の実施例における「プログラム復号・実行」処理1206の手順詳細を示す。

処理2001：　配布サーバ100から受信した暗号化済みプログラム152、もしくは基本プログラムデータ1801から取得した正規プログラム501のいずれかから、利用者固有命令コード列601、もしくは正規命令コード列503を取得する。取得した命令コード列を、以後の処理により実行していく。

処理2002：　これから実行する命令コード列が利用者固有命令コード列601であれば、処理2003に進む。正規命令コード列503であれば、処理2004に進む。

処理2003：　後述の処理2006において利用するテーブルとして、利用者固有ジャンプテーブル128を指定する。

処理2004：　後述の処理2006において利用するテーブルとして、正規ジャンプテーブル127を指定する。

処理2005：　処理2001もしくは処理2008によって取得した命令コード列から、次に実行する命令コードを取得する。

処理2006：　取得した命令コードを処理するルーチンのメモリアドレスを、各命令処理ルーチンへのジャンプテーブルから取得する。このときに参照するジャンプテーブルは、処理2003もしくは処理2004において指定された、正規ジャンプテーブル127もしくは利用者固有ジャンプテーブル128のいずれかである。前述の実施例と同様に、利用者固有命令を正規命令に戻す処理と、該当する処理ルーチンを取得する処理を、同時に行なっている。

処理2007：　取得したメモリアドレスを用いて、処理ルーチンを呼び出す。もし、処理対象命令が他のプログラムに対する関数呼びだし、もしくは関数呼びだしからのリターンなど、他の命令コード列に移動する処理であれば、処理2008を呼び出す。

処理2008：　関数呼びだしなどの命令コード列変更の処理を行なう。もし必要であれば正規プログラム501、暗号化済みプログラム152から命令コード列を取得する。その後処理2002に進む。

処理2009：　処理2005において取得した命令コードがプログラム実行終了を示すものであれば、プログラム実行終了処理ルーチンが呼び出され、処理が終了する。

処理2010：　処理2005において取得した命令コードが他の命令コード列へ移動する命令、プログラム実行終了命令のいずれでもないときは、当該命令の処理を行なった後に処理2005に進む。

【００６２】上記に示したように、二つ目の実施例では、暗号化されたプログラムと暗号化されていないプログラムを混在させて実行することが可能である。

【００６３】

【発明の効果】以上に述べたように、本発明による暗号

化されたプログラム実行方法によって以下の効果が得られる。
（１）第三者がプログラムを入手したときに、プログラムの実行を防ぐことができる。
（２）プログラム復号に必要な時間を大きく短縮可能である。
（３）復号のための記憶領域を必要としない。
（４）暗号化されたプログラムと、暗号化されていないプログラムを同時に実行可能である。

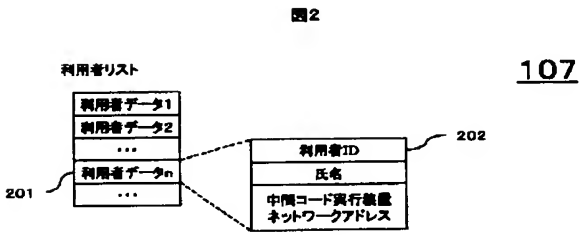
【図面の簡単な説明】

【図１】本発明を実現する実施例における、システム構成を示す図である。
【図２】利用者リストのデータ構成を示す図である。
【図３】中間コードの正規命令コード対応を示す図である。
【図４】命令コード対応リストのデータ構成を示す図である。
【図５】正規プログラムデータのデータ構成を示す図である。
【図６】暗号化済みプログラムのデータ構成を示す図である。
【図７】正規ジャンプテーブルのデータ構成を示す図である。
【図８】利用者固有ジャンプテーブルのデータ構成を示す図である。
【図９】プログラム配布要求のデータ構成を示す図である。
【図１０】本発明の中心となる、プログラム暗号化処理とプログラム解釈実行処理を示す図である。
【図１１】命令コード対応の生成・配布の処理手順を示す図である。
＊【図１２】プログラム配布・実行の処理手順を示す図である。
【図１３】命令コード対応生成の処理手順を示す図である。
【図１４】利用者固有命令コード対応生成の処理手順を示す図である。
【図１５】命令コード対応受信の処理手順を示す図である。
【図１６】プログラム暗号化の処理手順を示す図である。
【図１７】本発明を実現する一つ目の実施例における、プログラム復号・実行の処理手順を示す図である。
【図１８】本発明を実現する二つ目の実施例における、中間コード実行装置の構成を示す図である。
【図１９】基本プログラムデータのデータ構成を示す図である。
【図２０】本発明を実現する二つ目の実施例における、暗号化プログラム復号・実行の処理手順を示す図である。

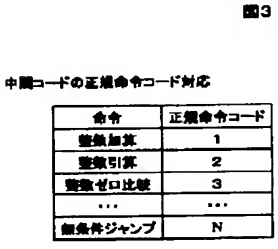
【符号の説明】

100…配布サーバ装置、101…主記憶装置、102…二次記憶装置、103…中央処理装置、104…通信装置、105…命令コード対応管理モジュール、106…プログラム管理モジュール、107…利用者リスト、108…命令コード対応リスト、109…正規プログラムデータ、120…中間コード実行装置、121…主記憶装置、122…通信装置、123…中央処理装置、124…プログラム管理モジュール、125…命令コード対応受信モジュール、126…プログラム復号・実行モジュール、127…正規ジャンプテーブル、128…利用者固有ジャンプテーブル、150…命令コード対応、151…暗号化済プログラム、152…プログラム配布要求

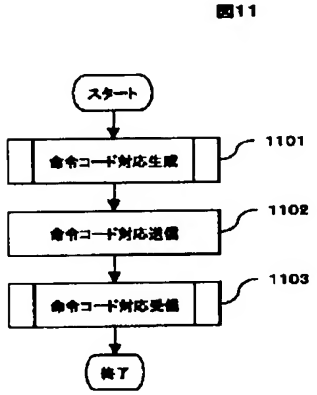
【図２】



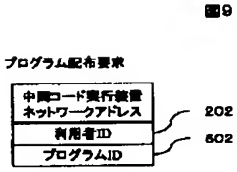
【図３】



【図１１】



【図９】



【図１】

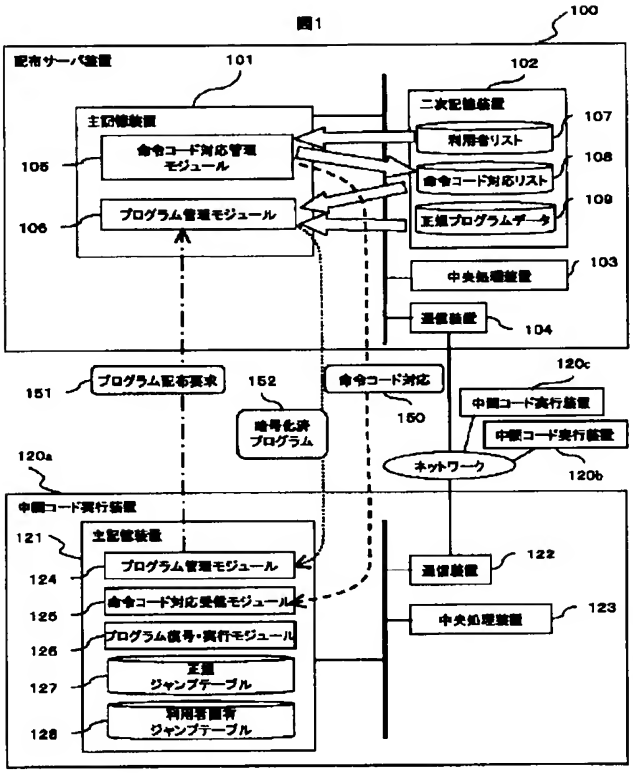


図1
配布サーバ装置
100
101
102
主記憶装置
二次記憶装置
命令コード対応管理モジュール
105
利用者リスト
107
命令コード対応リスト
108
プログラム管理モジュール
106
正規プログラムデータ
109
中央処理装置
103
通信装置
104
プログラム配布要求
151
152
暗号化済プログラム
命令コード対応
150
120c
中間コード実行装置
中継コード実行装置
ネットワーク
120b
120a
中間コード実行装置
121
主記憶装置
プログラム管理モジュール
124
命令コード対応受信モジュール
125
プログラム復号・実行モジュール
126
正規ジャンプテーブル
127
利用者固有ジャンプテーブル
128
通信装置
122
中央処理装置
123

【図４】

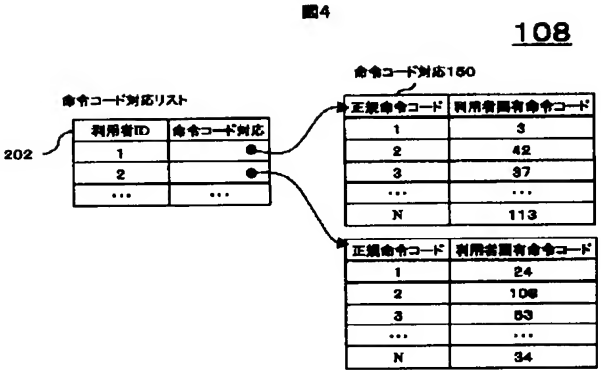


図4
108
命令コード対応150
命令コード対応リスト
202
利用者ID
命令コード対応
正規命令コード
利用者固有命令コード
1
3
2
42
3
37
N
113
正規命令コード
利用者固有命令コード
1
24
2
108
3
53
N
34

【図７】

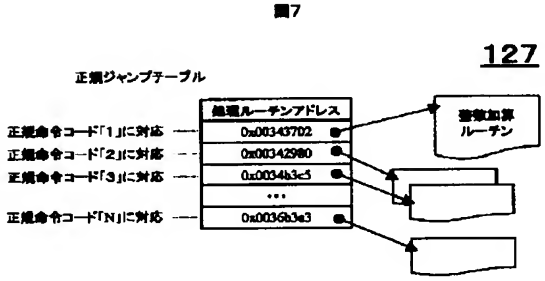


図7
127
正規ジャンプテーブル
処理ルーチンアドレス
正規命令コード「1」に対応
0x00343702
正規命令コード「2」に対応
0x00342980
正規命令コード「3」に対応
0x0034b3c5
正規命令コード「N」に対応
0x0036b3e3
登録加算ルーチン

【図５】

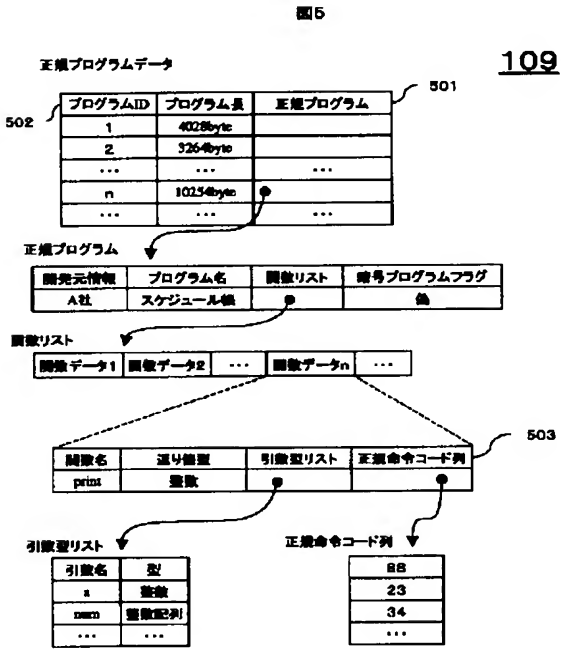


図5
正規プログラムデータ
109
501
502
プログラムID
プログラム長
正規プログラム
1
4028byte
2
326 4byte
n
10254byte
正規プログラム
開発元情報
プログラム名
関数リスト
暗号プログラムフラグ
A社
スケジュール帳
偽
関数データ1
関数データ2
関数データn
503
関数名
返り値型
引数型リスト
正規命令コード列
print
整数
引数型リスト
引数名
型
a
整数
num
整数配列
正規命令コード列
88
23
34

【図６】

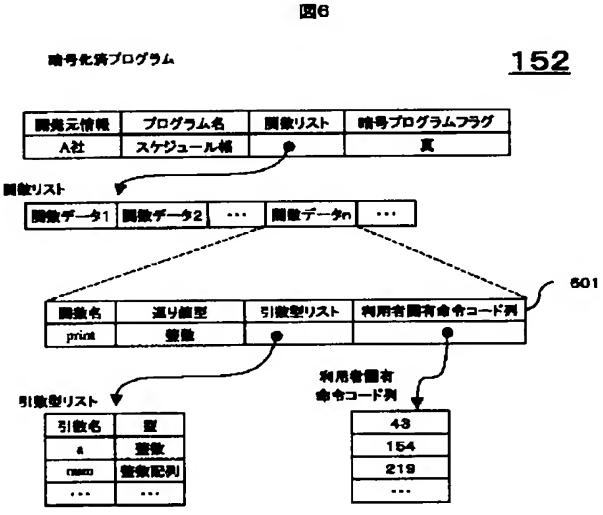
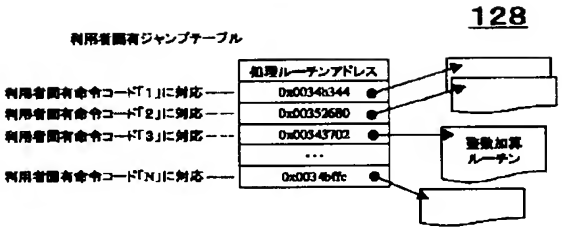


図6
暗号化済プログラム
152
開発元情報
プログラム名
関数リスト
暗号プログラムフラグ
A社
スケジュール帳
真
関数リスト
関数データ1
関数データ2
関数データn
501
関数名
返り値型
引数型リスト
利用者固有命令コード列
print
整数
引数型リスト
引数名
型
a
整数
num
整数配列
利用者固有命令コード列
43
154
219

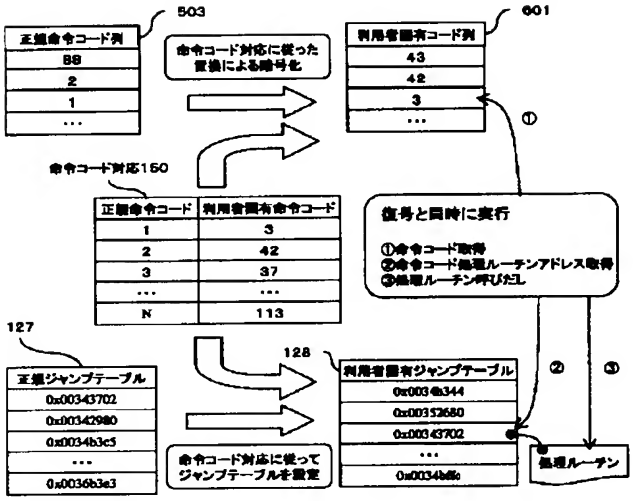
【図８】

図8



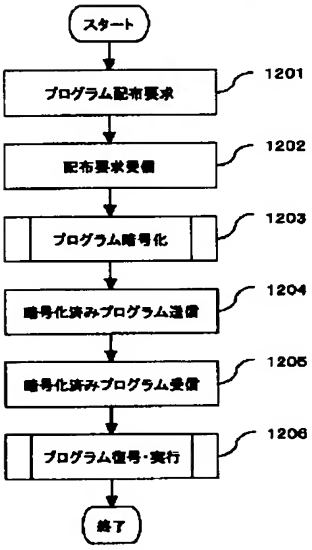
【図１０】

図10



【図１２】

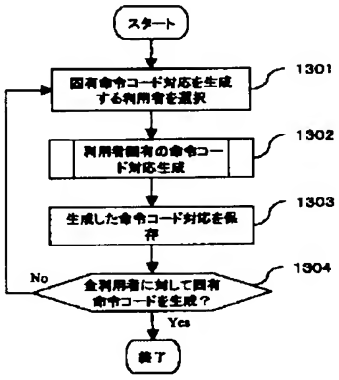
図12



【図１３】

図13

1101



【図１９】

図19

1801

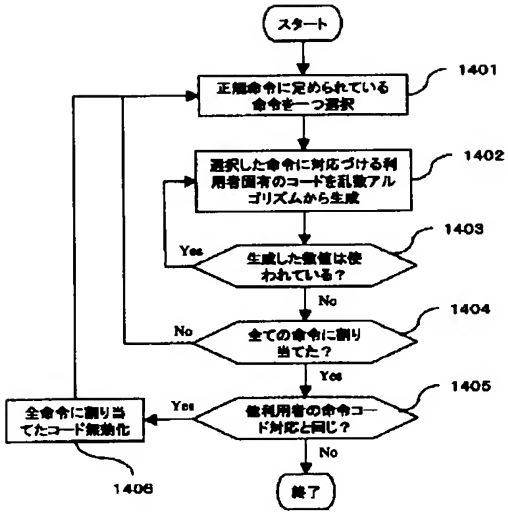
基本プログラムデータ

501

プログラムID	プログラム長	正規プログラム
1	1942byte	
2	469byte	
…	…	…
n	259byte	
…	…	…

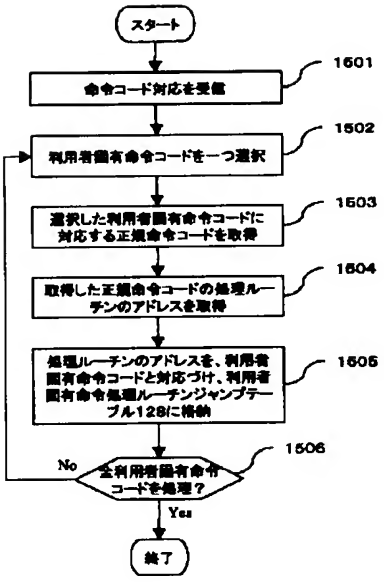
【図１４】

図14



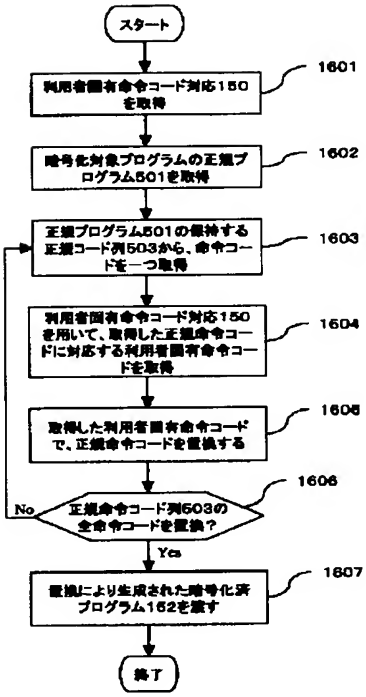
【図１５】

図15



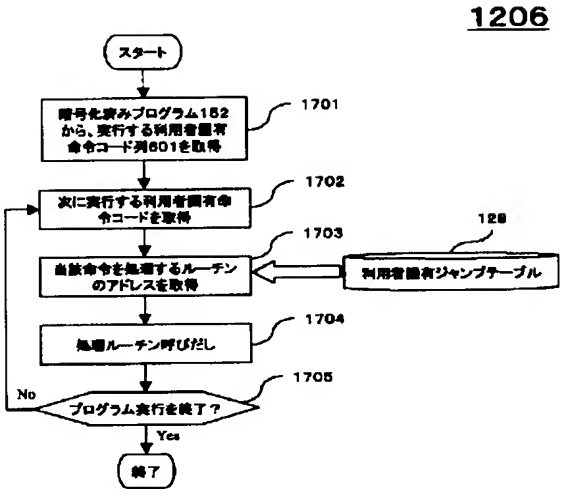
【図１６】

図16

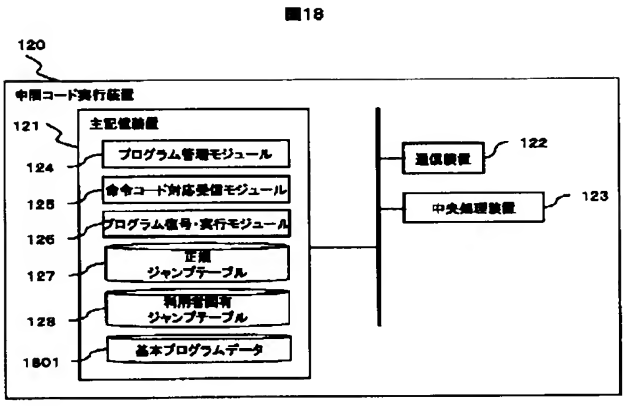


【図１７】

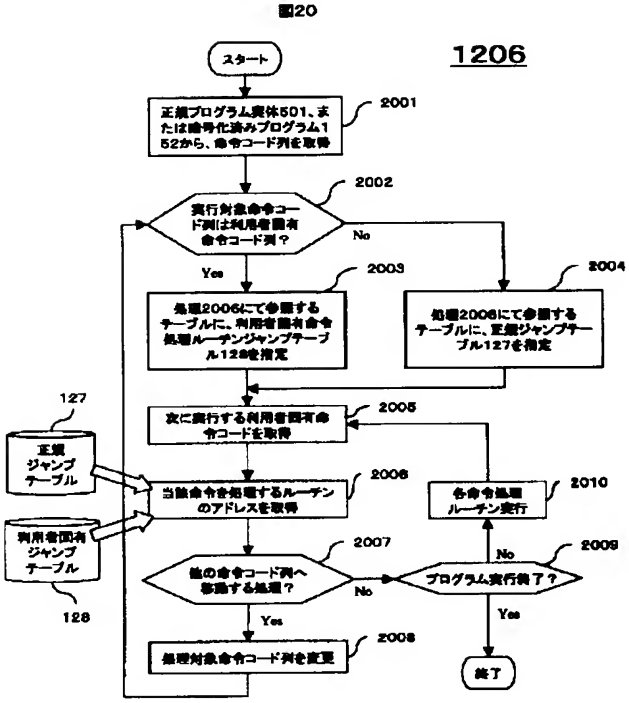
図17



【図１８】



【図２０】



フロントページの続き

(72)発明者　横山　泰子
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　森本　義章
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　北川　健二
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

Ｆターム(参考)　5B076 FA08 FA11